



ARCHITECTURE STYLE COMPARISON FOR NETWORK-BASED SOFTWARE IN THE SPHERE OF DISTANCE EDUCATION

Posudevskiy K.V.

LLC Playtika Bel, Belarussian State University of Informatics and Radioelectronics, Minsk, Belarus,
konstantin.posudevskiy@gmail.com

Abstract. This publication provides representative comparison of hierarchical architecture styles for the network-based software in the sphere of distance education. The comparison is represented in a table with further mentioned description. You can use the results of the comparison as ones to think about and sketch out your potential architecture.

As Roy Fielding mentioned, an architectural style is a coordinated set of architectural constraints that restricts the roles/features of architectural elements and the allowed relationships among those elements within any architecture that conforms to that style [1].

A software architecture is defined by a configuration of architectural elements – components, connectors, and data – constrained in their relationships in order to achieve a desired set of architectural properties [1].

Comprehensive information about architecture styles you can find in the Roy's dissertation.

Architecture style comparison can be based on comparison of its advantages and disadvantages, provided by architectural properties. Therefore, significant architectural properties have been determined for the network-based software in the sphere of distance education: user-perceived performance (UP Perform.), network performance (Net Perform), efficiency, scalability, simplicity, evolvability, reusability, visibility, portability, reliability.

Although, above mentioned architectural properties are common, its comprehensive description you can find in the Roy's dissertation.

I use a table of style versus architectural properties as the primary visualization for this classification. The table values indicate the relative influence that the style for a given row has on a column's property. Minus (–) symbols accumulate for negative influences and plus (+) symbols for positive, with plus-minus (±) indicating that it depends on some aspect of the problem domain (sphere of distance education). Although this is a gross simplification of the details presented in each section, it does indicate the degree to which a style has addressed (or ignored) an architectural property.

Further mentioned architecture styles are used for the comparison: client-server (CS), layered system (LS), layered-client-server (LCS), client-stateless-server (CSS), client-cache-stateless-server (C\$SS), layered-client-cache-stateless-server (LC\$SS), remote session (RS), remote data access (RDA).

Common information about the above mentioned styles you can find in the Roy's dissertation. A variety of client-server systems are surveyed by Sinha [2]. Layered systems and layered-client-server are described in details in the Umar's literature [3]. Advantages and disadvantages of the surveyed styles are based on Roy's dissertation classification of architecture styles for network-based applications [1].

The comparison of the hierarchal architecture styles is shown in table 1.

Table 1 – Evaluation of hierarchical styles for network-based software

Style	Net Perform.	UP Perform.	Efficiency	Scalability	Simplicity	Evolvability	Reusability	Visibility	Portability	Reliability
CS				+	+	+				
LS		–		+		+	+		+	
LCS		–		++	+	++	+		+	
CSS	–			++	+	+		+		+
C\$SS	–	+	+	++	+	+		+		+
LC\$SS	–	±	+	+++	++	++	+	+	+	+
RS			+	–	+	+		–		
RDA			+	–	–			+		–

The comparison classification uses relative changes in the architectural properties induced by each style as a means of illustrating the effect of each architectural style when applied to a system for sphere of distance education.

The set of architectural styles included in the comparison is by no means comprehensive of all possible network-based application styles. Indeed, a new style can be formed merely by adding an architectural constraint to any one of the styles surveyed.

The purpose of building software it is to create a system that meets or exceeds the application needs. The architectural styles chosen for a system's design must conform to those needs, not the other way around. Therefore, in order to provide useful design guidance, a classification of architectural styles should be based on the architectural properties induced by those styles [1]. You can use the results of the comparison as a simple classification of architectural styles for the software in the sphere of distance education.

Literature

1. Fielding, R. Th. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation. – University of California, Irvine. – 2000. – <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.
2. A. Sinha. Client-server computing. – Communications of the ACM, Volume 35 Issue 7. – July 1992. – 77-98 pp.
3. A. Umar. Object-Oriented Client/Server Internet Environments. – Prentice Hall PTR. – 1997.